

IN THE CLAIMS:

Please amend the claims as follows:

Claim 1 (Currently Amended): A method for displaying a desktop display surface having dimensions, comprising:

creating a render target surface having substantially the same dimensions as the desktop display surface;

casting the desktop display surface as a texture having the same dimensions as the desktop display surface;

determining a set of vertices that define a two dimensional rectangular object having the same dimensions as the desktop display surface;

rendering the two dimensional rectangular object by mapping at least a portion of the desktop display surface texture to the two dimensional rectangular object;

receiving a zoom factor, an offset in an x direction and an offset in a y direction;

calculating a texture addressing extent configured to determine the portion of the desktop display surface texture to be mapped to the two dimensional rectangular object;

calculating a set of texture addressing offsets in the x and y directions configured to provide the position on the desktop display surface texture from which the desktop display surface texture is to be mapped to the two dimensional rectangular object, wherein the texture addressing offset in the x direction is calculated as the offset in the x direction divided by the dimension of the desktop display surface in the x direction and the texture addressing offset in the y direction is calculated as the offset in the y direction divided by the dimension of the desktop display surface in the y direction; and

setting the render target surface as a scanout read location in preparation for displaying the desktop display surface.

Claim 2 (Canceled)

Claim 3 (Previously Presented): The method of claim 1, further comprising storing the rendered two dimensional rectangular object to the render target surface.

Claim 4 (Previously Presented): The method of claim 1, further comprising:
storing the rendered two dimensional rectangular object to the render target surface; and
scanning out the rendered two dimensional rectangular object from the render target surface.

Claim 5 (Previously Presented): The method of claim 1, further comprising:
receiving a zoom factor and one or more offsets; and
creating the two dimensional rectangular object according to the dimensions of the desktop display surface, the zoom factor and the one or more offsets.

Claim 6 (Previously Presented): The method of claim 1, further comprising:
receiving a zoom factor and one or more offsets; and
calculating the texture addressing coordinates (u, v) of each vertex in the set of vertices as a function of the dimensions of the desktop display surface, the zoom factor and the offsets.

Claim 7 (Previously Presented): The method of claim 1, further comprising:
receiving a zoom factor and one or more offsets; and
calculating a texture addressing extent configured to determine the portion of the desktop display surface texture to be mapped to the two dimensional rectangular object.

Claim 8 (Original): The method of claim 7, wherein the texture addressing extent is equal to a texture addressing range divided by the zoom factor.

Claim 9 (Canceled):

Claim 10 (Canceled):

Claim 11 (Currently Amended): The method of claim ~~[[9]]~~1, wherein creating the two dimensional rectangular object comprises:

setting a texture addressing coordinate u for an upper left hand corner of the two dimensional rectangular object to be equal to the texture addressing offset in the x direction; and

setting a texture addressing coordinate v for the upper left hand corner of the two dimensional rectangular object to be equal to the texture addressing offset in the y direction.

Claim 12 (Currently Amended): The method of claim [[9]]1, wherein creating the two dimensional rectangular object comprises:

setting a texture addressing coordinate u for an upper right hand corner of the two dimensional rectangular object to be equal to the texture addressing offset in the x direction plus the texture addressing extent; and

setting a texture addressing coordinate v for the upper right hand corner of the two dimensional rectangular object to be equal to the texture addressing offset in the y direction.

Claim 13 (Currently Amended): The method of claim [[9]]1, wherein creating the two dimensional rectangular object comprises:

setting a texture addressing coordinate u for a bottom left hand corner of the two dimensional rectangular object to be equal to the texture addressing offset in the x direction; and

setting a texture addressing coordinate v for the bottom left hand corner of the two dimensional rectangular object to be equal to the texture addressing offset in the y direction plus the texture addressing extent.

Claim 14 (Currently Amended): The method of claim [[9]]1, wherein creating the two dimensional rectangular object comprises:

setting a texture addressing coordinate u for a bottom right hand corner of the two dimensional rectangular object to be equal to the texture addressing offset in the x direction plus the texture addressing extent; and

setting a texture addressing coordinate v for the bottom right hand corner of the two dimensional rectangular object to be equal to the texture addressing offset in the y direction plus the texture addressing extent.

Claim 15 (Currently Amended): A method for displaying a desktop display surface, comprising:

receiving a zoom factor, an offset in an x direction and an offset in a y direction on the desktop display surface;

creating a two dimensional rectangular object having dimensions equal to dimensions of the desktop display surface;

computing a set of texture addressing coordinates for the two dimensional rectangular object using the dimensions of the desktop display surface, zoom factor, the offset in the x direction and the offset in the y direction;

casting a desktop display surface as a texture having dimensions equal to dimensions of the desktop display surface; and

rendering the two dimensional rectangular object by mapping at least a portion of the desktop display surface texture to the two dimensional rectangular object; and

calculating a texture addressing extent configured to determine the portion of the desktop display surface texture to be mapped to the two dimensional rectangular object; and

calculating a set of texture addressing offsets in the x and y directions configured to provide the position on the desktop display surface texture from which the desktop display surface texture is to be mapped to the two dimensional rectangular object, wherein the texture addressing offset in the x direction is calculated as the offset in the x direction divided by the dimension of the desktop display surface in the x direction and the texture addressing offset in the y direction is calculated as the offset in the y direction divided by the dimension of the desktop display surface in the y direction.

Claim 16 (Original): The method of claim 15, further comprising storing the rendered two dimensional rectangular object to a render target surface having substantially the same dimensions as the desktop display surface.

Claim 17 (Original): The method of claim 16, further comprising scanning out the rendered two dimensional rectangular object from the render target surface.

Claim 18 (Previously Presented): The method of claim 15, wherein the set of texture addressing coordinates includes texture addressing coordinates (u, v) for an upper right

hand corner, an upper left hand corner, a bottom left hand corner and a bottom right hand corner of the two dimensional rectangular object.

Claim 19 (Previously Presented): The method of claim 15, wherein computing a set of texture addressing coordinates for the two dimensional rectangular object comprises:

setting a texture addressing coordinate u for an upper left hand corner of the two dimensional rectangular object to be equal to the offset in the x direction divided by the dimension of the desktop display surface in the x direction; and

setting a texture addressing coordinate v for the upper left hand corner of the two dimensional rectangular object to be equal to the offset in the y direction divided by the dimension of the desktop display surface in the y direction.

Claim 20 (Previously Presented): The method of claim 15, wherein computing a set of texture addressing coordinates for the two dimensional rectangular object comprises:

setting a texture addressing coordinate u for an upper right hand corner of the two dimensional rectangular object to be equal to the offset in the x direction divided by the dimension of the desktop display surface in the x direction plus the inverse of the zoom factor; and

setting a texture addressing coordinate v for the upper right hand corner of the two dimensional rectangular object to be equal to the offset in the y direction divided by the dimension of the desktop display surface in the y direction.

Claim 21 (Previously Presented): The method of claim 15, wherein computing a set of texture addressing coordinates for the two dimensional rectangular object comprises:

setting a texture addressing coordinate u for a bottom left hand corner of the two dimensional rectangular object to be equal to the offset in the x direction divided by the dimension of the desktop display surface in the x direction; and

setting a texture addressing coordinate v for the bottom left hand corner of the two dimensional rectangular object to be equal to the offset in the y direction divided by the dimension of the desktop display surface in the y direction plus the inverse of the zoom factor.

Claim 22 (Previously Presented): The method of claim 15, wherein computing a set of texture addressing coordinates for the two dimensional rectangular object comprises:

setting a texture addressing coordinate u for a bottom right hand corner of the two dimensional rectangular object to be equal to the offset in the x direction divided by the dimension of the desktop display surface in the x direction plus the inverse of the zoom factor; and

setting a texture addressing coordinate v for the bottom left hand corner of the two dimensional rectangular object to be equal to the offset in the y direction divided by the dimension of the desktop display surface in the y direction plus the inverse of the zoom factor.

Claim 23 (Currently Amended): A computer system, comprising:

a processor; and

a memory comprising program instructions executable by the processor to:

create a render target surface having substantially the same dimensions as the desktop display surface;

cast the desktop display surface as a texture having the same dimensions as the desktop display surface;

determine a set of vertices that define a two dimensional rectangular object having the same dimensions as the desktop display surface;

render the two dimensional rectangular object by mapping at least a portion of the desktop display surface texture to the two dimensional rectangular object;

calculate a texture addressing extent configured to determine the portion of the desktop display surface texture to be mapped to the two dimensional rectangular object;

calculate a set of texture addressing offsets in the x and y directions configured to provide the position on the desktop display surface texture from which the desktop display surface texture is to be mapped to the two dimensional rectangular object, wherein the texture addressing offset in the x direction is calculated as the offset in the x direction divided by the dimension of the desktop display surface in the x direction and the texture addressing offset in the y

Supplemental Reply to Office Action of October 18, 2005

Atty. Dkt. No. NVDA/P001173

direction is calculated as the offset in the y direction divided by the dimension of the desktop display surface in the y direction; and

set the render target surface as a scanout read location in preparation for displaying the desktop display surface.

Claim 24 (Canceled)

Claim 25 (Original): The method of claim 23, wherein the memory further comprises program instructions executable to:

store the rendered two dimensional rectangular object to the render target surface; and

scan out the rendered two dimensional rectangular object from the render target surface.

Claim 26 (Previously Presented): The method of claim 7, wherein the texture addressing extent is equal to a dimension of the desktop display surface divided by the zoom factor.